UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/804,484	03/18/2004	Michael A. Rothman	42P18667	7590

7590 01/10/2007

Anthony H. Azure
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Seventh Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025

EXAMINER	
ABRAHAM, ESAW T	
ART UNIT	PAPER NUMBER
2133	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	01/10/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/804,484	ROTHMAN ET AL.	
	Examiner	Art Unit	
	Esaw T. Abraham	2133	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 March 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-28* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 March 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims **1-28** are presented for examination.

Specification

2. The specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim objections

3. Claims 11, 12, 26 and 27 are objected to because of the following informalities:

Claims 11 and 26 recite, "the debugger executable " since "executable" only suggests or makes optional, the term "executable" fails to further limit the claim. The examiner suggests ---the debugger executed---

Claims 12 and 27 recite, "the debugger operable " since "operable" only suggests or makes optional, the term "operable" fails to further limit the claim. The examiner suggests ---the debugger configured to operate---

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U. S. C 112

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 13 and 20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 13 is a hybrid claim. This is because the preamble of the claim is a system (an article of manufacture comprising a medium) but the body of the claim is a method step for initializing, entering and examining a debugger.

Claim 20 is a hybrid claim. This is because the preamble of the claim is a system (system comprising a computer, a processor and a flash memory) but the body of the claim is a method for of initializing, entering and examining a debugger.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere* CO., 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

5. Claims **1-28** are rejected under 35 U.S.C. 103(a) as being unpatentable over Barnstijn et al. (hereinafter " Barnstijn ") (U.S. PN: 5,715,387) in view of Lee (U.S. PN: 6,219,828).

As per claims 1 and 13:

Barnstijn teaches or discloses a system and a method associated with the fields of application debugging or program debugging. (see col. 1, lines 10-15). Barnstijn teach a host system (a first computer) is connected to the target system or remote system (second system) via a communications link. A program is loaded onto the host computer that translates operating system calls into a number of communication signals that are transmitted over this then loaded onto the target system. The target operating system in turn sends an operating system signal to a debugging application in the target system. The debugging application in the target system then sends an operating system signal to the operating system. The operating system translates the operating system signal to an event signal and sends the event signal (debug event or command) over the communications link and the host system receives the event signal from the communications link and processes the event signal using a communication program resident on the host system, thereby translating the event signal into a host system operating system signal (the first computer system communicatively coupled to the second communication system). This operating system signal is transferred to the application under development in the host system. In this manner both input and output events are physically executed or initiated using the target system's hardware while one or more applications under development reside and are executed in the host computer development environment (see col. 2, lines 46-67 and col. 3, lines 1-28). Barnstijn **does not explicitly teach** initializing a debugger during the pre-boot from a firmware environment. **However,** Lee in an analogous art teaches controlling a data processing system after startup, but before control is passed to the main operating

system and in particular to use of an Open Firmware user interface as a debugging tool (see col. 1, lines 7-10). Further, Lee teaches that the combination of pre-loaded software into ROM is termed "firmware". Boot firmware controls the computer from startup until control is handed over to the primary operating system (see col. 4, lines 1-9). **Therefore,** it would have been obvious to a person having an ordinary skill in the art at the time the invention was made to initialize a debugger during the pre-boot from a firmware environment as taught by Lee. **This modification** would have been obvious because a person having ordinary skill in the art would have been motivated in order to maintain a firmware debugger in the system that helps to resolve errors when testing newly developed Boot firmware Forth code (See col. 2, lines 53-57).

As per claims 2-4:

Barnstijn teaches that the target operating system in turn sends an operating system signal to a debugging application in the target system. The debugging application in the target system then sends an operating system signal to the operating system. The operating system translates the operating system signal to an event signal and sends the event signal (debug event or command) over the communications link and the host system receives the event signal from the communications link and processes the event signal using a communication program resident on the host system, thereby translating the event signal into a host system operating system signal (the first computer system communicatively coupled to the second communication system) (see col. 2, lines 46-67 and col. 3, lines 1-28).

As per claims 5-7:

Barnstijn teaches that the target operating system in turn sends an operating system signal to a debugging application in the target system. The debugging application in the target system then sends an operating system signal to the operating system. The operating system translates the operating system signal to an event signal and sends the event signal (debug event or command or wake-event) over the communications link and the host system receives the event signal from the communications link and processes the event signal using a communication program resident on the host system, thereby translating the event signal into a host system operating system signal (the first computer system communicatively coupled to the second communication system) (see col. 2, lines 46-67 and col. 3, lines 1-28).

As per claims 8 and 14:

Lee teaches that Kernel code, within the image, sets up an executing environment for the debugger, such as system exception handlers and debug console environment for the debugger, such as system exception firmware configuration variables are retrieved from a memory (see col. 3, lines 9-15).

As per claims 9 and 10:

Barnstijn teaches a host system (a first computer) is connected to the target system or remote system (second system) via a communications link and further a program is loaded onto the host computer that translates operating system calls into a number of communication signals that are transmitted over this then loaded onto the target system. The target operating system in turn sends an operating system signal to a debugging application in the target system (see col. 2, lines 58-67).

As per claims 11 and 12:

Lee teaches controlling a data processing system after startup, but before control is passed to the main operating system and in particular to use of an Open Firmware user interface as a debugging tool (see col. 1, lines 7-10). Further, Lee teaches that the combination of pre-loaded software into ROM is termed "firmware". Boot firmware controls the computer from startup until control is handed over to the primary operating system (see col. 4, lines 1-9).

As per claim 15:

Barnstijn teaches that the target operating system in turn sends an operating system signal to a debugging application in the target system. The debugging application in the target system then sends an operating system signal to the operating system. The operating system translates the operating system signal to an event signal and sends the event signal (debug event or command or wake-event) over the communications link and the host system receives the event signal from the communications link and processes the event signal using a communication program resident on the host system, thereby translating the event signal into a host system operating system signal (the first computer system communicatively coupled to the second communication system) (see col. 2, lines 46-67 and col. 3, lines 1-28).

As per claim 16:

Barnstijn teaches that the target operating system in turn sends an operating system signal to a debugging application in the target system. The debugging application in the target system then sends an operating system signal to the operating

system. The operating system translates the operating system signal to an event signal and sends the event signal (debug event or command) over the communications link and the host system receives the event signal from the communications link and processes the event signal using a communication program resident on the host system, thereby translating the event signal into a host system operating system signal (the first computer system communicatively coupled to the second communication system) (see col. 2, lines 46-67 and col. 3, lines 1-28).

As per claims 17-19:

Lee teaches controlling a data processing system after startup, but before control is passed to the main operating system and in particular to use of an Open Firmware user interface as a debugging tool (see col. 1, lines 7-10). Further, Lee teaches that the combination of pre-loaded software into ROM is termed "firmware". Boot firmware controls the computer from startup until control is handed over to the primary operating system (see col. 4, lines 1-9).

As per claim 20:

Barnstijn teaches or discloses a system and a method associated with the fields of application debugging or program debugging. (see col. 1, lines 10-15). Barnstijn teach a host system (a first computer) is connected to the target system or remote system (second system) via a communications link. A program is loaded onto the host computer that translates operating system calls into a number of communication signals that are transmitted over this then loaded onto the target system. The target operating system in turn sends an operating system signal to a debugging application in the target

system. The debugging application in the target system then sends an operating system signal to the operating system. The operating system translates the operating system signal to an event signal and sends the event signal (debug event or command) over the communications link and the host system receives the event signal from the communications link and processes the event signal using a communication program resident on the host system, thereby translating the event signal into a host system operating system signal (the first computer system communicatively coupled to the second communication system). This operating system signal is transferred to the application under development in the host system. In this manner both input and output events are physically executed or initiated using the target system's hardware while one or more applications under development reside and are executed in the host computer development environment (see col. 2, lines 46-67 and col. 3, lines 1-28). Barnstijn **does not explicitly teach** initializing a debugger during the pre-boot from a firmware environment. **However,** Lee in an analogous art teaches controlling a data processing system after startup, but before control is passed to the main operating system and in particular to use of an Open Firmware user interface as a debugging tool (see col. 1, lines 7-10). Further, Lee teaches that the combination of pre-loaded software into ROM is termed "firmware". Boot firmware controls the computer from startup until control is handed over to the primary operating system (see col. 4, lines 1-9). **Therefore,** it would have been obvious to a person having an ordinary skill in the art at the time the invention was made to initialize a debugger during the pre-boot from a firmware environment as taught by Lee. **This modification** would have been obvious because a

person having ordinary skill in the art would have been motivated in order to maintain a firmware debugger in the system that helps to resolve errors when testing newly developed Boot firmware Forth code (See col. 2, lines 53-57). Further, Barnstin in view of Lee **do not explicitly** teach or detail the first computer comprising a processor coupled to a memory. **Nevertheless**, as would have been well known to one ordinary skill in the art at the time the invention was made, a processors and a memory are required for computers in order to perform any operations and store the operation results. **Accordingly,** it would have been obvious to one ordinary skill in the art to include a processor and a memory because such components would have been required in order to perform computer operations and store the operation either temporarily or permanently in the computer.

As per claim 21:

Lee teaches that Kernel code, within the image, sets up an executing environment for the debugger, such as system exception handlers and debug console environment for the debugger, such as system exception firmware configuration variables are retrieved from a memory (see col. 3, lines 9-15).

As per claim 22:

Barnstijn teaches that the target operating system in turn sends an operating system signal to a debugging application in the target system. The debugging application in the target system then sends an operating system signal to the operating system. The operating system translates the operating system signal to an event signal and sends the event signal (debug event or command) over the communications link

and the host system receives the event signal from the communications link and processes the event signal using a communication program resident on the host system, thereby translating the event signal into a host system operating system signal (the first computer system communicatively coupled to the second communication system) (see col. 2, lines 46-67 and col. 3, lines 1-28).

As per claims 23-24:

Barnstijn teaches that the target operating system in turn sends an operating system signal to a debugging application in the target system. The debugging application in the target system then sends an operating system signal to the operating system. The operating system translates the operating system signal to an event signal and sends the event signal (debug event or command or wake-event) over the communications link and the host system receives the event signal from the communications link and processes the event signal using a communication program resident on the host system, thereby translating the event signal into a host system operating system signal (the first computer system communicatively coupled to the second communication system) (see col. 2, lines 46-67 and col. 3, lines 1-28).

As per claim 25:

Barnstijn teaches a host system (a first computer) is connected to the target system or remote system (second system) via a communications link and further a program is loaded onto the host computer that translates operating system calls into a number of communication signals that are transmitted over this then loaded onto the

target system. The target operating system in turn sends an operating system signal to a debugging application in the target system (see col. 2, lines 58-67).

As per claims 26-28:

Lee teaches controlling a data processing system after startup, but before control is passed to the main operating system and in particular to use of an Open Firmware user interface as a debugging tool (see col. 1, lines 7-10). Further, Lee teaches that the combination of pre-loaded software into ROM is termed "firmware". Boot firmware controls the computer from startup until control is handed over to the primary operating system (see col. 4, lines 1-9).

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

US PN: 6,219,782 Khan et al.

US PN: 5,850,562 Crump et al.

7. Any inquiry concerning this communication or earlier communication from the examiner should be directed to Esaw Abraham whose telephone number is (571) 272-3812. The examiner can normally be reached on M-F 8-5.

If attempts to reach the examiner by telephone are successful, the examiner's supervisor, Albert DeCady can be reached on (571) 272-3819. The fax phone numbers for the organization where this application or proceeding is assigned are (571) 273-8300 for regular communications and (571) 273-8300 for after final communications.

Information regarding the status of an Application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or PUBLIC PAIR. Status information for unpublished applications is available through Private Pair only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Esaw Abraham

Esaw Abraham

Art unit: 2133

ALBERT DECADY
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100